

「コンピュータアーキテクチャ（改訂 5 版）」第 2 刷 変更箇所

ページ	1 刷	2 刷
p.4 上から 8 行目	バロース	バロース(Burroughs)
p.4 下から 6 行目	主記憶を共用しつつ,	主記憶を共用するため,
p.66 下から 4 行目	実アドレスは, …	物理アドレスは, … 【本書全体で実アドレス→物理アドレスに変更】
p. 70 上から 8 行目	仮想記憶	仮想記憶(3・4・4 項参照)
p. 103 下から 7～9 行目	2 つの命令は, まったく異なるレジスタを使用しているため,	命令①の読み出すレジスタ r1, r2 及び書き込むレジスタ r3 と, 命令②の読み出すレジスタ r4, r5 及び書き込むレジスタ r6 とは, まったく異なるレジスタであるため,
p.104 上から 13 行目	タスク	プロセス
p.150 下から 10～12 行目	メガバイト	ギガバイト(GB) 【図 3・78 に合わせて、メガバイトからギガバイトに修正】
p.157 図 3・85		(提供 キオクシア(株))を追記 【提供元を明記】
p.163 下から 10～12 行目	タスク	プロセス 【用語の変更(2 箇所)】
p.175 上から 6 行目	後述の	3・2・6 項で述べた
p.175 上から 14～16 行目	したがって, (a-1) から(a-4) の利点から, 性能を重視する場合には命令キャッシュとデータキャッシュを分けるとともに, (d-1), (d-2) の問題点に対してはコンパイラやハードウェア機構の工夫により対処する.	したがって, 次項で述べる 1 次キャッシュにおいては, (a-1) から(a-4) の利点から性能を重視してセパレートキャッシュとし, 2 次キャッシュ以降では, (d-2) の利用効率を重視してユニファイドキャッシュとするのが一般的である.
p.182 下から 1 行目	メモリデータ	メモリデータレジスタ
p.185 下から 2 行目	先の	図 3・85 に示した
p.186 下から 1 行目	実記憶(主記憶)	主記憶
p.187 上から 2～3 行目	仮想記憶の管理は, ハードウェアと OS との組み合わせにより, ユーザとは全く独立に行われる. この結果,	以降, プログラムが使用するアドレスを仮想アドレス(virtual address)(あるいは論理アドレス(logical address)), 主記憶上のアドレスを物理アドレス(physical address)(あるいは実アドレス(real address))と呼ぶ. また, 仮想アドレスから物理アドレスへのアドレスマッピングを管理するシステムを仮想記憶管理システムと呼ぶ. このような仮想記憶によって,
p.189 上から 12 行目	(iii) …	(iii)全体を削除
p.189 上から 15 行目	なお, 以下の議論ではページング方式を…	なお, 以下の議論では, マルチプログラミング方式を前提とする. また, ページング方式を…
p.192 下から 9 行目	タスク	プロセス

p.194 8行目から p.199 6行目	実アドレス 実アドレスキャッシュ (real address cache)	物理アドレス 物理アドレスキャッシュ (physical address cache)
p. 194 図 3・115 図注 (2 か所)	アドレス空間 ID(コンテキストの ID) 仮想マシン ID	アドレス空間 ID(VMID によって特定される仮想マシン上のコンテキスト ID) 【アドレス空間 ID と VMID の関係を明記】 仮想マシン ID(1 つの物理プロセッサ上で動作する複数 OS の ID) 【仮想マシン ID の意味を補足】
p. 207 図 3・122		FibreChannel 中の「分散*1」を性能欄から調停欄に移動
p. 211 下から 12 行目	一フィルタ	フィルタ 【一を削除】
p. 215 上から 11～12 行目	基づくもので、ディスクにおけるダウンサイジング(1・4 節参照)である。	基づくものである。
p.220 上から 4 行目	光磁気ディスク(magneto-optic disk, MO)が主流となっている。	光磁気ディスク(magneto-optic disk, MO)が使用されている。
p.220 下から 8～10 行目	同様の使い方をされるフレキシブルディスクと比べ、MO の記録層の保持力は 1 桁以上大きく、しかも透明基板の内側に保護されているため、温度、湿度等外部環境の変化にも強いという特長をもつ。	【削除】
p.222 上から 6～8 行目	この場合には、I/O の能力 < CPU の能力であり、I/O バウンド(I/O bound)と呼ぶ。そうでない場合には、CPU の使用率 $u_{CPU} = 1$ で、CPU バウンドとなる。	図 3・136 の場合には、 $u_{CPU} < 1$ で、CPU にアイドル状態があり、I/O バウンド(I/O bound)と呼ぶ。もし、 $t_{CPU} + t_{I/O}$ の間に m 未満のプログラムしか走れない場合には、CPU の使用率 $u_{CPU} = 1$ で、CPU バウンド(CPU bound)となる。
p.222 【例題 3・27】とその解 差替え	【例題 3・27】図 3・135 において、 $n = 100$, $m = 4$, $t_{CPU} = 1\text{ms}$, $t_{I/O} = 50\text{ms}$, $t_z = 1\text{s}$ とする。これは I/O バウンドか CPU バウンドか。 (解) $mt_{CPU} = 4 \times 1 \leq t_{CPU} + t_{I/O} = 1 + 50$ より、これは I/O バウンドである。	【例題 3・27】図 3・136 で、① $m = 8$, $t_{CPU} = 1\text{ms}$, $t_{I/O} = 20\text{ms}$, ② $m = 8$, $t_{CPU} = 2\text{ms}$, $t_{I/O} = 10\text{ms}$ の各場合に I/O バウンドか CPU バウンドか。 (解) ① $mt_{CPU}(= 8) < t_{CPU} + t_{I/O}(= 1 + 20)$ より I/O バウンド。② $mt_{CPU}(= 16) > t_{CPU} + t_{I/O}(= 2 + 10)$ より CPU バウンド。
p.226 下から 7～8 行目	一般にメッセージサイズはまちまちであるがメッセージによって、転送に要する時間やバッファ領域の所要量が大きくばらつく可能性があり、	一般にメッセージサイズはまちまちであるため、転送に要する時間やバッファ領域の所要量がメッセージによって大きくばらつく可能性があり、
p.245 上から 3～4 行目	…ある場合に起きる。依存関係は…	…ある場合に起きる。ここで参照とは変数の値を読むことであり、定義とは書き込むことである。依存関係は…

p.245 下から 6～8 行目	例えば、図 4・8(a) に示す命令列において、加算命令の実行結果は普通なら WB まで進まないといほかの命令で参照することができない。したがって、これを命令デコーダで検出し、2 ステージ分実行を遅らせる必要がある。	例えば、図 4・8(a) で i0 と i1 は R3 によってフロー依存関係にある。i0 の実行結果は WB まで進まないといほかの命令で参照することができないため、依存関係を命令デコーダで検出し、i1 の実行を 2 ステージ遅らせる必要がある。
p.245 下から 1 行目	挿入に対応して	挿入によって
p.246 上から 5 行目	R2 中の変数 c に関する	R2 に関する
p.249 下から 8 行目	共有するもの	共用するもの
p.256 【例題 4・4】	(2) 基本ブロック内でのレジスタ変数の依存関係を抽出せよ。 (解) 基本ブロックは①～④、⑤～⑬の 2 つである。レジスタ変数の依存関係を図 4・21 に示す。なお、図中の矢印の表記法は、先に述べた(2) データハザードでの定義に従っている。	(2) 先に述べたパイプラインハザード中の(2)データハザードの項(p.245)で定義した依存関係とその表記法を用いて、基本ブロック内でのレジスタ変数の依存関係を抽出し、図示せよ。 (解) (1) 基本ブロックは①～④、⑤～⑬の 2 つである。 (2) レジスタ変数の依存関係を表記法に従って図 4・21 に示す。
p.257 図 4・21		図中の「li」の位置を⑫の近くに移動
p.258 上から 2 行目	基本ブロック 2 中の破線は	基本ブロック中の破線は
p.299 図 4・55	[42]	[70] 【文献番号変更】
p.304 下から 2 行目	更新され、	更新された場合に、
p.322 下から 12～14 行目	あるプロセッサから共有メモリへの書込みの際には、同一アドレスに対するすべての読み出しと書き込みが完了していることを条件とする。一方、読み出しに際しては、同一アドレスに対するすべての読み出しが完了していればよい。	あるプロセッサから共有メモリへの書込みの際には、すべての読み出しと書き込みが完了していることを条件とする。一方、読み出しに際しては、すべての読み出しが完了していればよい。
p.323 上から 13～14 行目	例えば、図 4・70 のプログラムにおいて、P1 において読み出した flag2 の値と P2 において読み出した flag1 の値を考える。シーケンシャルコンシステンシのもとでは、プログラムオーダが守られ、かつそれぞれのメモリ操作が完了してから次のメモリ操作が行われるため、これらの値が同時に 0 になることはなく、相互排除が実現される。	以上のリラクストコンシステンシモデルのもとで、図 4・69 のプログラムを実行することを考えてみる。
p.323 上から 1 行目 と 4 行目の冒頭		【字下げ】
p.347 図 4・90		図中の左側線上に 2 つ並ぶ「σ」の右の 1 つを、右側線上の同位置に移動。
p.347 下から 9 行目	クロス網においては、 $m \geq n$ ならクロスバと同じ非閉塞網となる。	クロス網は、 $m \geq 2n-1$ なら非閉塞網、 $m \geq n$ のとき再構成型非閉塞網となる。

p.347 下から 5～7 行目	～を抑えている.	～を抑えている. クロス網は, n, m, r の設定により柔軟な構成が可能であり, 図 4・91 のベネス網も中央の破線部を2つの $Clos(2; 2; 2)$ とすると, 全体は $Clos(2; 2; 4)$ となる.
p.347 下から 1～2 行目	クロス網と異なり, ファットトリーでは, 段と段の間が双方向の通信となる. このため, 第一段の入出力数は共に $n + m$ となる.	ファットトリーでは, 段の間が双方向通信となり, 一段目は r 個の $(n + m) \times (n + m)$ スイッチとなる.
p.348 図 4・92 (b)		「 n 入力」の直下に「 n 出力」を追記
p.388 文献[70] 【差し替え】	M. Kobayashi, and M.H. MacDougall: The Stack Growth Function: Cache Line Reference Models, IEEE Trans. on Computers, Vol.C-38, No.6, pp.798-804(1989).	R.W. Hockney, and C.R. Jesshope: Parallel Computers 2, IOP Publishing Ltd, p.625(1988)

(ISBN 978-4-274-22615-1) オーム社